

linux认证:在Linux中利用变量给文件命名Linux认证考试 PDF
转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/585/2021_2022_linux_E8_AE_A4_E8_AF_c103_585091.htm 前不久有一个Linux系统管理员向笔者求助。他说他们在Linux操作系统上部署了一个Oracle 10G的数据库系统。现在他们利用数据库系统中的数据泵工具从系统中备份数据。他们希望每天把这个备份文件复制到一个特定的地方。到这一步为止都没有问题。但是他们希望在复制的过程中，能够利用变量名对这个备份文件进行命名。如按星期几的不同，分别命名为1backup.dmp,2backup.dmp等等。前面的1、2就表示星期几的意思。这么设置的话，一个星期一个轮回。到下个星期一的话，新的备份文件就会把旧的备份文件替换掉。如此的话，备份文件所占用的硬盘空间也就不会无限制的增加上去。笔者对于他们为什么不直接采用Oracle提供的备份工具制定备份策略对数据库进行备份感到不解。不过要实现上面这个利用变量来给文件命名的话，还是可以实现的。笔者根据他们企业的需求，给这个系统管理员详细的阐述了实现方法。

一、文件创建时间与系统时间的关系。在利用变量来命名文件名字的时候(如时间变量)，需要注意一个问题，就是文件的创建时间与系统时间的关系。如上面这个需求，如果某个文件是在星期一创建的，而在星期二时复制这个文件。那么此时文件的名字应该为多少呢？是按文件的创建时间来命名，还是按复制的时间来命名的。如果从上面这个需求来看，利用文件的创建时间来命名更加的合理。因为这个创建时间才真正体现了数据库的备份时间。

二、相关命令参数解析。如果要把一个文件复制到另外一

个地方，同时利用时间变量对其进行重命名的话，那么可以利用如下的命令来实现`cp mydb.log "(date %w)"mydb.log`。这个命令的意思就是把mydb.log复制到一个特定的位置(当前目录下)，并对此进行重新命名。命名的规则就是在原来的文件变量前面加上一个时间参数，这里采用的是星期的时间变量。`"(date %w)"`这个参数就是截取星期几的时间变量。注意，这里的时间是按照文件的创建时间来的，而不是参考文件复制的时间。也就是说，如果这个文件的创建时间是星期三，而复制时间是星期四的话，那么这个变量的值就为3，而不是4。因为文件的创建时间为3。笔者以前刚开始接触Linux操作系统的时候，经常犯这个错误。希望笔者这个过来人常犯的错误，能够引起大家的警惕。另外上面这个时间参数，必须要用双引号括起来。不过不这么做的话，那么系统就会提示这个命令有错误。这是一个语法上的错误。所以如果系统管理员在批处理程序中使用这个命令的话，那么最好能够进行预先测试。因为这些语法错误的话，就算再老到的系统管理员，一不小心就容易犯。实践是检验真理的唯一标准，这句话是不会错的。还有一个小细节就是`"(date %w)"`与`"(date %w)"`有区别吗?这个既有区别，又可以说没有。这主要是看用在什么场合了。如果用在CP命令中，则中间加不加()`加号`，是一样的。但是，在其他一些命令场合中，必须要加入中间这个加号。否则的话，系统会提示错误。如下面笔者要谈到的ECHO命令，就有这方面的要求。三、利用Echo命令来进行测试。如果系统管理员对于自己编写的命令参数不怎么肯定的话，那么就可以利用echo命令来进行测试。Echo命令可以显示环境变量的值，也可以显示某些特定变量的值。其实

这些变量就是系统中的环境变量。如下图所示。利用echo \$(date %w)就可以显示当天系统时间。这个命令只显示星期几。如果使用这个命令的话，跟cp命令中最终的时间参数还是有一点差距，主要体现在以下这几个方面。首先，要利用echo命令来显示系统变量值的时候，必须在变量前面加上\$符号。如果加上这个符号，系统就会认为这是一个环境变量，或者由环境变量转换过来的一个变量。如果不加这个符号的话，则操作系统会无法识别，也就不能够正常显示这个变量的值了。其次，在这个变量中，中间必须加入号。如上图所示，如果不加入这个号的话，则操作系统就会提示错误，说%w是一个无效的参数。但是在cp命令中，则没有这个强制的限制。在中间加不加这个加号，都能够达到同样的效果。不过为了提高参数的准确性，最好大家还是按照这个echo命令可以认可的格式下。因为echo命令可以认可的格式，则在其他命令中是通用的。相反，其他命令中可以用的格式，在echo等命令上不一定通用。所以为了提高编写的脚本程序的移植性，最好能够采用通用的变量书写方式。第三，在cp命令中如果使用时间变量的话，在必须使用双引号括起来。否则的话，系统会无法识别变量名称。而在echo命令中，则不能够使用双引号。因为如果使用双引号的话，则操作系统会认为这是一个常量，而不是一个环境变量。如上图所示，如果在echo命令中采用了双引号的话，则其直接显示的是双引号中的内容。而不会把双引号中的内容当作环境变量。这系统管理员在书写的时候，需要注意这个格式上的细微区别。第四，需要强调的是，在Linux操作系统的脚本程序，跟Windows操作系统中的脚本程序与命令行程序不同的是，

前者对于脚本程序中的命令、参数都是区分大小写的。而在Windows操作系统中则是不区分大小写。所以在Linux操作系统中编写脚本程序或者使用命令变量的时候，一定要注意大小写是敏感的。通常情况下，Linux操作系统下所采用的变量、参数都是小写的。特别是那些以前从事Windows操作系统的维护工作，而现在使用Linux操作系统，在这个转型的过程中，需要注意这方面的差异。

四、其他需要提醒的注意点

。 `cp mydb.log $(date %w)mydb.log`，这个命令与上面的命令比较类似，但是这里笔者用的是\$符号。虽然两者命令看起来比较类似，但是功能上有比较大的差异。在这个地方，其虽然也是利用星期几作为变量名，但是其参考的却是系统的时间，而不是文件创建的时间。即如果是星期四创建的文件，而在星期五进行复制作业，则其重命名的文件是5mydb.log。可见如果要采用的是系统的时间来作为参考变量的话，仍然需要采用\$这个环境变量提示符。另外就是权限方面的问题。如采用非root用户创建了一个sh文件(这相当于Windows操作系统下的bat文件)，默认情况下这个文件的所有权人是无法执行这个文件的，因为其没有这个权限。为此系统管理员需要利用root帐户登陆，然后重新调整权限。如果不怎么熟悉调整权限的话，那么最简单的就是把所有权限赋予给所有的人。就`chmod 777`就可以完成了。不过如果采用的是其他的脚本程序的话，如定期删除某个文件，则会留下一定的安全隐患。为此在权限设计的时候，不能够贪图省力。还是应该遵循最小权限的原则，只给有需要的用户赋予其完成必要使命的最小权限。如这里所示，如果脚本编写完成，那么只需要给这个用户执行这个脚本程序的权限即可。系统管理员甚至可

以不把这个脚本程序修改或者删除的权限给这个用户。防止其意外修改或者删除了这个脚本文件。更多优质资料尽在百考试题论坛 百考试题在线题库 linux认证更多详细资料 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com